III. REMARKS

Claims 1-2, 5-16, 18-32, 34-43 and 45-50 are pending in this application. By this Amendment, claims 1, 13, 27, 31 and 40 - 42 have been amended. The above amendments and the following remarks are being made to facilitate early allowance of the presently claimed subject matter. Applicants do not acquiesce in the correctness of the rejections and reserve the right to present specific arguments regarding any rejected claims not specifically addressed. Further, Applicants reserve the right to pursue the full scope of the subject matter of the original claims in a subsequent patent application that claims priority to the instant application. Reconsideration in view of the above amendments and the following remarks is respectfully requested.

Entry of this Amendment is proper under 37 C.F.R. §1.116(b) because the Amendment: (a) places the application in condition for allowance as discussed below; (b) does not raise any new issues requiring further search and/or consideration; and (c) places the application in better form for appeal. Accordingly, Applicants respectfully request entry of this Amendment.

In the Office Action, claims 1-2, 6, 8, 10-16, 19, 21, 23-25, 27-32, 35-38, 40-43 and 46-49 were rejected under 35 U.S.C. § 102(b) as anticipated by OraRep (Oracle 7 Server Distributed Systems, Vol. II: Replicated Data, Release 7.3, February, 1996, Oracle Corporation); claims 5, 18, 34, 39, 45 and 50 are rejected under 35 U.S.C. § 103(a) as being unpatentable over OraRep in view of Pal et al. (USPN 6,598,079); and claims 7, 9, 20, 22 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over OraRep in view of OraAdm (Oracle 8i Administrator's Reference, Release 3 for Sun SPARC Solaris, ORACLE CORPORATION, August 2000). Withdrawal of these rejections is respectfully requested, for the reasons that follow:

1. OraRep does not disclose each and every claimed feature.

I-1 Differences between OraRep and the current invention

The claimed invention recites, *inter alia*, "repeating the steps of logging at least one transaction and executing the at least one logged transaction on the second server prior to the step of queuing ***until a set point is met***[,]" and that a "***time duration of each repeating [update] step is shorter than a preceding repeating step***, and ***transaction service on the second sever is paused*** until the providing step[,]" as recited in claim 1 and claimed similarly in claims 13, 27, 31 and 40-42. (Emphasis ours). As described by the specification of the current invention, an execution of a logged transaction on the target server, i.e., an update, occurs more quickly than the execution of the same transaction on the source server since the target server is not interacting with users ("transaction service on the second (target) server [being] paused until the providing step" (claim 1 of the current invention)). Therefore, if an update is repeated , a duration of time for each of the later updates will be shorter than a prior update. In addition, each update handles less transactions, which further shortens the duration of time required. For example, if copying a database on the target server takes 8 hours, the copy of the database on the target server is 8 hours behind the database on the source server. An update of the transactions that occurs in the 8 hours on the source server will take less than 8 hours on the target server, e.g., 6 hours, because there is less data and no interaction between the target server and users. After the update, which takes 6 hours, the database on the target server is 6 hours behind the database on the source server, and a second update is needed to make up this 6-hour lag. The time for the second update will be less than 6 hours for the same reason. Therefore, each later update will be for a reduced duration of time than a prior update. The current invention repeats this process until a set point is met, e.g., the database on the target server is acceptably identical to that on the source server, and then transfers all users to the target server, i.e., it conducts a database migration. The target server provides transaction service only after its database is acceptably identical to that on the source server, i.e., "transaction service on the second server is paused until the providing step." Claim 1 of the current invention (similarly claimed in claims 13, 27, 31 and 40 - 42).

Applicants respectfully submit that OraRep does not disclose, *inter alia*, the above identified features of the current invention. OraRep discloses a method of replicating a database from a source server to a target server, which presents a number of fundamental differences compared to the claimed database migration. In particular, OraRep aims to transfer the content of one database to another database periodically despite the two databases ***being accessed by two different user groups at the same time***. In contrast, during a database migration according to the current invention, the two databases are close to identical or identical for an instance of time prior to transferring users from the first to the second server. During a database migration, "***transaction service on the second server is paused*** until [the database migration is completed]." Claim 1 of the current invention (similarly claimed in claims 13, 27, 31 and 40-42). (Emphasis ours). In OraRep, the target server provides transaction service during replication, so it cannot achieve a "time duration of each repeating [update] step [being] shorter than a preceding repeating step," as in the current invention. Claim 1 of the current invention (similarly claimed in claims 13, 27, 31 and 40-42).

The following example illustrates the difference between OraRep and the current invention. Suppose we have database 1 (D1) and database 2 (D2), and data is to be transferred from D1 to D2, and time1 (T1) is the beginning time for the transfer and time 2 (T2) is the end time. In OraRep, D1 begins transferring data to D2 at T1, and at T2, D2 has what D1 had at T1. OraRep does not update D2 with D1's new records obtained after T1. By sharp contrast, in the current invention, D1 begins transferring data to D2 at T1 and repeats logging and executing transactions for D2. As a result, at T2, D2 is substantially identical to D1 at T2. To achieve the substantially identical D2, in the current invention, repeating logging and executing are needed until a set standard is met for the identicalness because during the first transfer, D1 might obtain additional data. OraRep does not try to make D1 and D2 identical at time T2 (end of replication), so there is no disclosure of repeating updates until a set point is met. In OraRep, between T1 to T2, there is only one transfer. In OraRep, D1 might transfer to D2 at a subsequent time, but such a transfer is not a ***repeated*** update because there is a preset time interval between the two transfers. *See,* e.g., OraRep 2-3. In OraRep, a subsequent transfer has no relationship with the prior transfer between T1 and T2 and is not capable of achieving a D2 substantially identical to D1 at T2.

In OraRep, the replication begins periodically at a fixed time interval (only applies to asynchronous mode) and the time duration of each replication, although not mentioned in OraRep, is dependent on the size of the data to be replicated and the work load of the two servers (***with both source and target servers providing transaction service***, as contrary to the current invention). That is, the time duration for a replication may increase or decrease depending on usage. As admitted by the Office in the Office Action dated 4/21/2004, OraRep does not discuss time durations relative to replications. In addition, because OraRep does not try to achieve a D2 that is substantially identical to D1 at time T2, OraRep does not need to set a point for stopping the update repeating.

1-2 Response to the Office's assertion

In the Office Action, the Office recites some teachings of OraRep to support the assertion that OraRep anticipates the current invention. Applicants respectfully submit that the Office's understanding of those OraRep teachings is erroneous. First, the Office randomly selects pieces of the OraRep teachings from different sections and combines those pieces of teachings in a manner that is inconsistent with the principles and the original teachings of OraRep. For example, in rejecting claims 1, 13, 27, 40 and 41, the office cites both the synchronous mode and the asynchronous mode of OraRep, such as:

> *at Page 4-33 where Figure 4-4 shows site B having database copy maintaining synchronously with the database at master definition site A;* and
> *at Page 4-28, Figure 4-3 where request of transaction, database replication execution at the remote site, is always queued before being propagated to the destination site[.]* -- Office Action at page 3

These teachings conflict. In OraRep, synchronous and asynchronous are two distinctive modes for update. OraRep 2-2. In the synchronous mode, changes in one site are updated to another site immediately, while in the asynchronous mode, changes in one site are stored and forwarded to another site periodically at a preset time interval. Id. In the synchronous mode, there is no deferred transaction queue. OraRep 4-33. In the asynchronous mode, there is no repeating of updating because there is a preset time interval between two replications. OraRep 2-2. In the asynchronous mode, "you should select an interval that is greater than the length of time required to perform a refresh." OraRep 3-17. Accordingly, either the synchronous mode and the asynchronous mode of OraRep lacks some features of the current invention. The Office erroneously includes the teachings of the synchronous mode and the asynchronous mode together in the rejection of the current invention. Applicants submit that the two modes are separate and cannot be combined.

Second, the Office incorrectly interprets OraRep. The Office concludes that OraRep discloses "repeating the steps of logging at least one transaction and executing the at least one logged transaction on the second server prior to the step of queuing until a set point is met" (claim 1 of the current invention). To support this conclusion, the Office asserts OraRep discloses that "the replicated transactions at the second server are repeated queued and applied when the set point job_queue_interval time is reached. ... Once a job execution starts, new jobs will be queued but will ***not be executed until the next scheduled time arrives.***" Office Action at page 4. (Emphasis ours). Correct interpretation of this asserted disclosure of OraRep is that the execution of newly queued transactions ***begins*** when the next scheduled time arrives. In contrast, in the current invention, the update repeating ***stops*** when a set point is met. It is incomprehensible how OraRep could anticipate such a different feature of the current invention, despite the Office's assertion.

Third, the Office asserted that OraRep discloses "a time duration of each repeating step is shorter than a preceding repeating step because of less system loading or no user activities." Office Action at page 13. Applicants respectfully traverse this assertion because OraRep does not disclose this feature and, in operation, only obtains such a result by chance. In OraRep, the targets are continuing to do transactions in addition to the replication. OraRep 2-4, ("if one site becomes unavailable, you cannot update any other replicas until the downed site either becomes available or is dropped from the replicated environment.") In OraRep, there is no disclosure that a target server has less system loading than the source server. Simply because a source server may transfer to a plurality of targets does not mean that the targets are less loaded than the source. That is, in OraRep, during the period of a replacement, the source can obtain more records than those transferred in the replacement and, work load of targets being the same, the time duration of the next replacement will be longer. Also, system loading of the targets in the later replacement can be heavier than in the preceding replacement, which also causes longer replacement duration in the later replacement. Accordingly, OraRep does not teach that "a time duration of each repeating step is shorter than a preceding repeating step because of less system loading or no user activities" (Office Action at page 13). OraRep may only obtain such a result by chance. In contrast, in the current invention, "a time duration of each repeating step is shorter than a preceding repeating step[,]" due to the features that "transaction service on the second server is paused until the providing step[,]" and that updates are repeated. Claim 1 (similarly claimed in claims 13, 27, 31 and 40-42).

What is more fatal to the Office's assertion is that OraRep does not make the target substantially identical to the source. In OraRep, there is no disclosure of "no user activities" and "less system loading" in the target. Office Action at page 13. In OraRep, there is no disclosure of "the time duration of each repeating [update] step [being] shorter than a preceding repeating step[,]" as recited in claim 1 of the current invention.

In view of the foregoing, Applicants respectfully request withdrawal of the rejections.

2. There is no motivation or suggestion to combine OraRep and Pal et al.

Applicants submit that there is no motivation or suggestion to combine OraRep and Pal et al. because the basic principles of OraRep and Pal et al. are different, and it is not feasible for OraRep to adopt the feature of Pal et al. regarding increasing or decreasing time durations.

Pal et al. disclose "a pledge-based resource allocation system[.]" Col. 1, line 53. The Pal et al. system "[allocates] resources to clients for a limited time period[,]" which is set by the system. Col. 1, lines 56-57. The limited time is set to "[ensure] that a client cannot allocate a resource for so long as to affect other client's use of the resource." Col. 1, lines 57-58.

The different principles of OraRep and Pal et al. make them incompatible. OraRep needs to ensure that all of the data that the resource server had at the beginning of the replication is transferred to the target server at the end of the replication, no matter how long the replication is. In contrast, Pal et al. do not ensure that the limited time allocated to a client is long enough for the client to complete its use of the resource. If one tries to combine OraRep and Pal et al., she/he will face the dilemma that either the allocated time is not enough for completing the replication or that the replication time is so long as to affect other client's use of the resource. The basic problem is that no one knows how long a replication in OraRep will be and thus any allocated time limit will be arbitrary. In view of the forgoing, Applicants submit that the Office has failed to show a suggestion or motivation to combine, either in OraRep, or in Pal et al., or in the knowledge generally available to one of ordinary skill in the art. Accordingly, Applicants request withdrawal of the rejections.

In the Office Action, the Office asserts:

> *the Pal reference is mainly utilized to provide the teaching of adjusting time*

> *duration based on the number of objects allocated for further combining with OraRep teaching on a time duration of each repeating step is shorter than a preceding repeating step[,]" so that "it would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Pal's reference with OraRep's by shortening intervals between executions of queued deferred remote procedure calls such that the number of jobs queued would be decreased steadily to an optimal level because of a time duration of each repeating step is shorter than a preceding repeating step, and furthermore, the two references are directed to job queue, resource allocation, time limit for allocation and time interval of transaction scheduling.*
>
> -- Office Action at page 14.

Applicants respectfully traverse this assertion. First, the two cited references are not both "directed to job queue, resource allocation, time limit for allocation and time interval of transaction scheduling." For example, as stated above, OraRep does not disclose a time limit for allocation and Pal et al. do not disclose a job queue. Neither OraRep nor Pal et al. disclose a time duration for transaction scheduling. Second, even if "the Pal reference is mainly utilized to provide the teaching of adjusting time duration based on the number of objects allocated" (Office Action at page 13), Pal et al. and OraRep cannot be combined because adjusting the allocated time duration in Pal et al. is still arbitrary and does not ensure that a replication in OraRep can be completed in the adjusted time duration. In addition, in OraRep, the time for completing a replication is also dependent upon the workload of the target server. Pal et al. provide no disclosure regarding this issue. That is, OraRep cannot achieve a decreased number of jobs by adopting Pal et al. In view of the foregoing, the OraRep system cannot adopt teachings of Pal et al. regarding assigning a time duration for replicating a database because the assigned time duration may be less than the time required for the replication.

In view of the foregoing, Applicants respectfully request withdrawal of the rejections.

In the Office Action, the Office asserts that the combination of OraRep and OraAdm makes claims 7, 9, 20, 22 and 26 obvious. Applicants respectfully traverse this assertion. Applicants submit that those dependent claims are allowable, *inter alia*, for their allowable base claims. Applicants reserve the right to provide further arguments regarding this rejection.

Claims 2 and 5-12 are dependent on claim 1, claims 14-16 and 18-26 are dependent on claim 13, claims 28-30 are dependent on claim 27, claims 32 and 34-39 are dependent on claim 31 and claim 44-50 are dependent on claim 42. The dependent claims are believed to be allowable based on the above arguments, as well as for their own additional features.

Applicants respectfully submit that the application is in condition for allowance. Should the Examiner believe that anything further is necessary to place the application in better condition for allowance, he is requested to contact Applicants' undersigned attorney at the telephone number listed below.

Respectfully submitted,

Spencer K. Warnick
Reg. No. 40,398

Date: 1/11/05

Hoffman, Warnick & D'Alessandro LLC
Three E-Comm Square
Albany, New York 12207
(518) 449-0044
(518) 449-0047 (fax)